

Detektion von Strukturelementen in Volumendaten

Diplomarbeit
im Institut für Medien- und Phototechnik
an der Fachhochschule Köln

Autor
Christian Uphoff
Friedberg
Matr.-Nr.: 11021689

1. Prüfer: Prof. Dr. Christian Blendl
2. Prüfer: Prof. Dr. Stefan Grünvogel

Köln, im Februar 2010

Detection of structural elements in volume data

Thesis
at the Department
of
Imaging Sciences and Media Technology
University of Applied Sciences Cologne

Author
Christian Uphoff
Friedberg
Mat.-No.: 11021689

First Reviewer: Prof. Dr. Christian Blendl
Second Reviewer: Prof. Dr. Stefan Grünvogel

Cologne, February 2010

Kurzbeschreibung

Titel: Detektion von Strukturelementen in Volumendaten

Autor: Christian Uphoff

Referenten: Prof. Dr. Christian Blendl und Prof. Dr. Stefan Grünvogel

Zusammenfassung:

Im Rahmen dieser Diplomarbeit wurde ein Java-Programm zur automatisierten Detektion von Strukturelementen in Volumendatensätzen entwickelt und anhand eines neu entwickelten Prüfkörpers der lokale geometrischen Fehler eines bildgebenden Systems ermittelt. Mit geeigneten Algorithmen werden, für eine treffsichere Detektion, das Rauschen reduziert, die Stützstellen detektiert und die 7-Parameter der Helmert-Transformation bestimmt.

Stichwörter: Digitale Radiologie, Prüfkörper, automatisierte Auswertung, Bildverarbeitung, Java, ImageJ, Detektion

Sperrvermerk: Die Einsicht in die vorliegende Arbeit ist bis zum 28. Februar 2012 gesperrt.

Datum: 25. Februar 2010

Abstract

Title: Detection of dedicated structures in volume data

Author: Christian Uphoff

Reviewers: Prof. Dr. Christian Blendl and Prof. Dr. Stefan Grünvogel

Summary:

The goal of this thesis is to create a software based on Java for automatical detection of structural elements in volume data. By using a newly developed phantom the local geometrical distortion of an imaging system can be measured. With appropriate algorithms the noise is reduced, the net points are detected and the 7 parameters of the Helmert-Transformation are calculated.

Keywords: Phantom, automated readout, Image Processing, Java, ImageJ, Detection

Closing remark: This Thesis is closed until 2012-02-28.

Date: 25th February, 2010

Inhaltsverzeichnis

1	Einleitung	1
1.1	Einführung.....	1
1.2	Aufgabenstellung	1
2	Grundlagen und Material	2
2.1	Der Prüfkörper	2
2.1.1	Prüfkörperverfahren	2
2.1.2	Der Prüfkörper	2
2.2	Verwendete Software und Klassen	4
2.2.1	Java	5
2.2.2	ImageJ	5
2.2.3	Eclipse IDE for Java Developers	6
2.2.4	3D Object Counter	6
2.2.5	3D Hybrid Median Filter	7
2.2.6	DICOM Format	9
2.3	7 – Parameter – Transformation	10
2.3.1	Die Helmert-Transformation	10
2.3.2	Methode nach Horn	12
2.3.3	Quaternionen	14
3	Programmaufbau und Implementierung der Algorithmen	15
3.1	Entwicklungsumgebung	15
3.2	Eingabedaten	16
3.2.1	Einlesen der Volumendaten	16
3.2.2	Einlesen der Referenzkoordinaten	17
3.3	Die Benutzeroberfläche	18
3.4	Bestimmung von Orientierung und Lage des Prüfkörpers	19
3.4.1	Detektion der Bildanker	20
3.4.2	Zuordnung der Bildanker	21
3.4.3	Erste Näherung der Transformationsparameter	23
3.5	Bestimmung der Transformationsparameter	24
3.5.1	Detektion der Stützstellen	25
3.5.2	Behandlung von falsch positiv Ergebnissen	25
3.5.3	Bestimmung des lokalen geometrischen Fehlers	26
3.6	Ausgabe der Ergebnisse	28

4	Ergebnisse	29
4.1	Rauschreduktion	29
4.2	Fehlerbetrachtung.....	31
4.2.1	Virtueller Datensatz	31
4.2.2	Reeller Datensatz	33
5	Diskussion	34
6	Zusammenfassung und Ausblick	36
A	Anhang	38
A.1	Abbildungsverzeichnis	38
A.2	Tabellenverzeichnis	39
A.3	Formelverzeichnis	39
A.4	Inhalt der CD	39
A.5	Literaturverzeichnis	40

1 Einleitung

1.1 Einführung

Ziel jeder radiotherapeutischen Behandlung ist es, die Tumorregion mit einer möglichst hohen Bestrahlungsdosis möglichst zielgenau zu belasten und gleichzeitig die benachbarten Risikoorgane bestmöglich zu schonen. Die dafür nötigen Dosisberechnungen basieren auf Planungsdaten die i.d.R. mit Computertomographie-Systemen gewonnen werden. Entspricht die morphologische Realität nicht den Planungsgrundlagen, wird eventuell die Tumorregion zu wenig oder ein Risikoorgan zu hoch belastet. Die digitale Volumen Tomographie (DVT) als relativ neues Verfahren für die volumetrische Darstellung des Schädels wird bislang nur für diagnostische Zwecke verwendet. Durch die Verbesserung der örtlichen Genauigkeit kann die DVT ebenfalls für die Bestrahlungsplanung verwendet werden.

Jedes radiologische bildgebende Verfahren ist fehlerbelastet. In der Radiographie ist die durch die Zentralprojektion bedingte Verzeichnung ein unabänderlicher Faktor. In der DVT kommen noch Verzerrungen durch die Algorithmen hinzu, die bei der Rekonstruktion der 2D Aufnahmen in 3D Bilddaten verwendet werden.

Es muss eine verlässliche und rekonstruierbare Methode gefunden werden, mit deren Hilfe sich die gerätespezifische Verzerrung bestimmen lässt.

1.2 Aufgabenstellung

Im Rahmen der Diplomarbeit soll ein Programm zur automatischen Detektion von Strukturelementen in einem Prüfkörper entwickelt werden, der mit der DVT erfasst wurde. Es wird ein geeignetes Koordinatensystem - Transformation benötigt, um die gewonnenen Daten mit den Daten eines Referenzkörpers vergleichen und die Abweichungen bestimmen zu können. Im Ergebnis soll sich ein drei dimensionales Vektorfeld ergeben, das den lokalen geometrischen Fehler des bildgebenden Systems an den Stützstellen in Richtung und Stärke beschreibt.

In den folgenden Kapiteln werden die Verfahren und Methoden dargelegt, die zur Erreichung des Arbeitsziels verwendet werden.

2 Grundlagen und Material

Im folgenden Kapitel werden der Prüfkörper, Software sowie technische und mathematische Grundlagen und Standards näher vorgestellt.

2.1 Der Prüfkörper

Um die möglichen gerätespezifischen Raumverzerrungen einer Digitalen Volumen Tomographie (DVT) vermessen zu können, bedarf es eines möglichst exakt zu konstruierenden und auswertbaren Prüfkörpers. Die Hauptkenngrößen des Prüfkörpers sind Anzahl und Verteilung der Stützstellen sowie die genaue Bestimmbarkeit ihrer Zentroide.

2.1.1 Prüfkörperverfahren

Bei der Prüfung von Qualitätsparametern bildgebender Systeme werden neben physikalischen Einzelmessungen häufig auch Prüfkörper verwendet. Diese Vorlagen können virtuell als Datensätze oder real als Körper vorliegen. Mit Datensätzen können Ausgabegeräte überprüft werden. Mit Körpern können Eingabegeräte und ganze Systeme kontrolliert werden. Das grundlegende Verfahren ist in allen Fällen das gleiche: die Vorlagen und die Ergebnisse werden verglichen, was Rückschlüsse auf das System oder die Komponenten zulässt.

Für die Röntgentechnik werden Prüfkörper in der Regel aus Plexiglas® (PMMA) oder solid water (sw) hergestellt. Das verwendete Material soll eine ähnliche Abbildung ergeben wie menschliches Muskel- oder Fettgewebe. Die Strukturen, deren Übertragung durch das System untersucht wird, sind aus Metall. Auch diese Strukturen sollen denen auf realen Aufnahmen ähnlich sein, z.B. ähnelt Aluminium in seinen Schwächungseigenschaften Knochen.

2.1.2 Der Prüfkörper

Der Prüfkörper ist aus mehreren soliden „Solid Water“ Kunststoffscheiben konstruiert und mit einer Toleranz von $\pm 0,1$ mm gefertigt. An den Kontaktflächen der Scheiben sind Aluminiumkugeln mit einem Durchmesser von 6mm eingelassen. Insgesamt ergibt sich eine Verteilung von 100 Aluminiumkugeln

auf 10 Ebenen in einem Kugelvolumen mit einem Durchmesser von 15cm und entspricht somit dem Volumen eines menschlichen Schädels. Im zentralen Bereich des Körpers ist zusätzlich für die Gewebedichtesimulation eine spezielle Torte montiert, die aus verschiedenen Materialien unterschiedlicher Elektronendichte besteht. Eine Auflistung der Materialien ist in Tabelle 2-1 nachzulesen.

Material	Formel
Luft	75%N, 23.2%O, 1.3%A
PMP	$[C_6H_{12}(CH_2)]$
Polystyrol	$[C_8H_8]$
PMMA	$[C_5H_8O_2]$
Aluminium	Al
Teflon	$[CF_2]$

Tabelle 2-1: Im Prüfkörper verbaute Materialien unterschiedlicher Elektronendichte.

Mit Ihrer Hilfe kann eine Aussage über den Zusammenhang von gerätespezifischem Grauwert des bildgebenden Systems, Massenschwächungskoeffizient und Elektronendichte des Materials getroffen werden. Diese Information ist im Besonderen für die Verwendung in der Bestrahlungsplanung von hoher Relevanz.

Der Schematische Aufbau und eine volumetrische Visualisierung der im DVT erfassten Bilddaten ist in Abbildung 2-1 dargestellt.

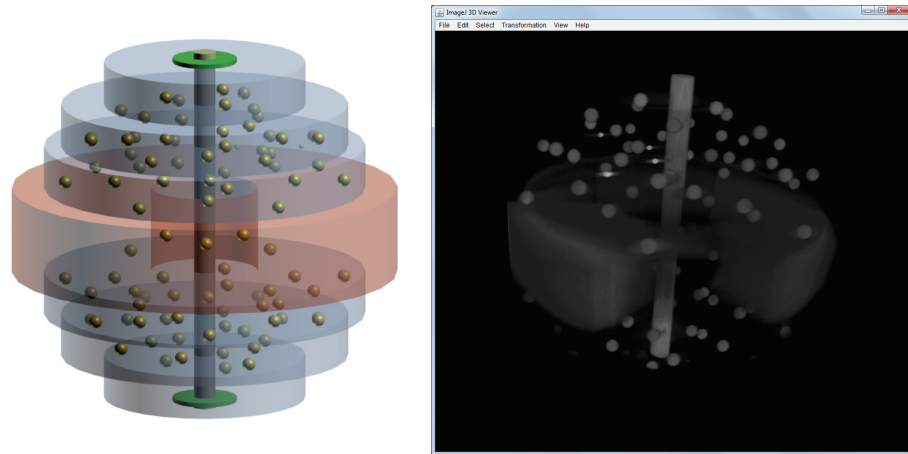


Abbildung 2–1: Schematischer Aufbau (links) und mit ImageJ realisierte Volumendarstellung der Daten des mit DVT erfassten Prüfkörpers (rechts)

Weitere Strukturelemente sind in diesem Prüfkörper enthalten, wie z.B. ein zentrales Al-Rohr, mit Hilfe dessen die Modulationstransferfunktion (MTF) in jeder Schicht im Zentrum der Schicht berechnet werden kann. Die Bestimmung von Größen, welche die Bildqualität eindeutig beschreiben wie die MTF oder das Rauschleistungsspektrum (W) ist aber nicht Aufgabe dieser Arbeit. Diese Arbeit beschäftigt sich ausschließlich mit der Bestimmung der räumlichen Unsicherheit bildgebender Systeme.

Um die relative Lage des Prüfkörpers im Raum automatisch bestimmen zu können, sind auf der vierten Ebene vier Stahlkugeln mit einem Durchmesser von 2 mm eingelassen. Diese Punkte werden im weiteren Bildanker (BA) genannt.

2.2 Verwendete Software und Klassen

Dies ist eine vollständige Liste der im Rahmen der vorliegenden Arbeit verwendeten Software. Alle Programme und Bibliotheken sind frei verfügbar und

können auf den angegebenen Seiten bezogen werden. Die folgenden Beschreibungen sind den jeweiligen Paketen größtenteils unverändert entnommen.

2.2.1 Java

Verwendung findet die Programmiersprache Java und die frei verfügbare Bildverarbeitungssoftware ImageJ, die voll in Java implementiert ist. Zudem bieten ihre Java-Bibliotheken bereits implementierte, häufig verwendete Algorithmen an. Als Entwicklungsumgebung wurde Eclipse verwendet. Es wird außerdem die Java Runtime benötigt.

Version: Verwendet wurde Version 1.6.0

Bezugsquelle: [http:// java.sun.com/](http://java.sun.com/)

2.2.2 ImageJ

ImageJ ist ein public domain Java Bildbearbeitungs-Programm, das in einer Java Virtual Machine der Version 1.4 oder höher auf einer Vielzahl verschiedener Plattformen läuft, unter anderem auf Windows, Mac OS, Mac OS X und Linux. Es beherrscht einige verschiedene Dateiformate und Farbtiefen, kann mit Bildstapeln umgehen und ist sehr leicht durch Plugins zu erweitern. Zusätzlich stehen bereits viele Plugins aus der medizinischen und biologischen Bildverarbeitung für ImageJ zur Verfügung.

Version: Verwendet wurde Version 1.43 in Kombination mit Java 1.6.0.

Bezugsquelle: <http://rsb.info.nih.gov/ij/>

2.2.3 Eclipse IDE for Java Developers

Eclipse ist ein quelloffenes Programmierwerkzeug zur Entwicklung von Software verschiedenster Art. Ursprünglich wurde Eclipse als integrierte Entwicklungsumgebung für die Programmiersprache Java genutzt, aufgrund seiner Erweiterbarkeit wird es aber mittlerweile auch für viele andere Entwicklungsaufgaben eingesetzt. Für Eclipse gibt es eine Vielzahl von Erweiterungen sowohl quelloffen als auch von kommerziellen Anbietern.

Version: Verwendet wurde Version 3.5.1

Bezugsquelle: <http://www.eclipse.org/>

2.2.4 3D Object Counter

Im Rahmen dieser Arbeit werden Klassen des ImageJ Plugins „3D Object Counter“ (Cordelières 2006) in der Version 2.0 zur Detektion der Kugeln und zur Bestimmung ihrer Zentroide verwendet. Diese erwiesen sich bei Tests mit Testdaten als sehr zuverlässig und robust. 3D Object Counter basiert auf einer Form des sogenannten Schwellwert-Verfahrens.

Das Schwellwert-Verfahren kann auch als Thresholding oder als Grauwertdiskriminierung bezeichnet werden. Bei dieser Methode wird die Zugehörigkeit eines Voxels zum Objekt oder zum Hintergrund anhand seines Grauwertes im Vergleich mit einem Schwellwert bestimmt. Ist der Grauwert des Voxel niedriger als der Schwellwert, so wird das Voxel je nach Anwendung dem Objekt oder dem Hintergrund zugeordnet, während Voxel mit einem höheren Grauwert als der Schwellwert der anderen Kategorie zugeordnet werden.

Die Erkennung und Zuordnung der Pixel zu Hintergrund und Objekten erfolgt in zwei Schritten:

In Abbildung 2-2 ist der erste Durchgang schematisch dargestellt. Als Vorlage dient ein Graustufenbild (a) das durch eine Schwellenwertoperation Binarisiert wird (b). Der Volumendatensatz wird von seiner oberen linken zu seiner unteren rechten Ecke verarbeitet. Wird der erste Objektpixel gefunden, wird ihm ein Index zugeordnet (c). Der Index ist eine eindeutige Nummer, welche jeder

Pixel trägt, das zum selben Objekt gehört. Jedes Mal, wenn ein neuer Objektpixel gefunden wird, werden seine 13 vorhergehenden Nachbarn (9 auf der vorhergehenden und 4 auf derselben Ebene) auf ihren Index überprüft. Ist ein Index vorhanden, wird dem neuen Objektpixel dieser zugeordnet (d-h) (Cordelières 2009).

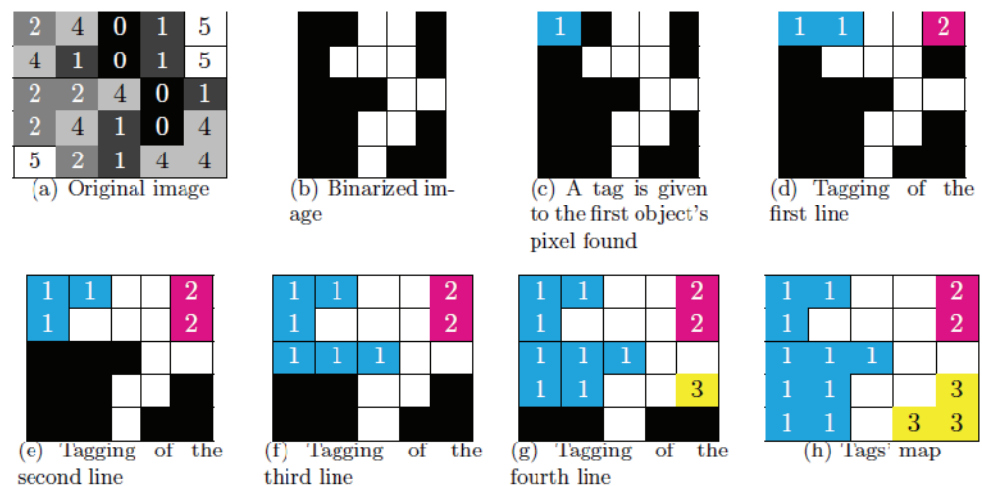


Abbildung 2-2: Verfahren der Objektzuordnung durch das ImageJ Plugin „3D Object Counter“

Quelle: Fabrice P. Cordelières, 2009

Im zweiten Durchgang werden Mehrdeutigkeiten aufgelöst. Wenn z.B. ein U-förmiges Objekt gefunden wird, werden seine beiden Arme als zwei unterschiedliche Objekte erkannt da sie nur am unteren Ende verbunden sind. Während des Regulierungsprozesses, werden benachbarte Objektpixel die unterschiedliche Indizes tragen erkannt und der höhere durch den niedrigeren Index ersetzt.

2.2.5 3D Hybrid Median Filter

Um auch bei niedrigen Signal – Rausch Verhältnissen eine zuverlässige Objekterkennung zu gewährleisten ist eine kantenerhaltende Rauschunterdrückung erforderlich. In das Programm wurde das ImageJ Plugin 3D Hybrid Median Filter (Mauer 2007) in einer modifizierten Form implementiert.

Grundsätzlich ist ein Median Filter eine Form der nicht-linearen oder auch Rangordnungs-Filter. Der Ergebniswert der Filter beruht nicht auf einer Linearkombination, sondern auf der Sortierung der Grauwerte der Umgebungspixel. Nicht-lineare Filter besitzen einen höheren Rauschabstand als lineare und Kanten werden weniger geglättet.

Bei der Medianfilterung werden die Grauwerte innerhalb einer Maske der Größe nach sortiert. Dann wird als neuer Grauwert für den betrachteten Pixel der mittlere Wert verwendet.

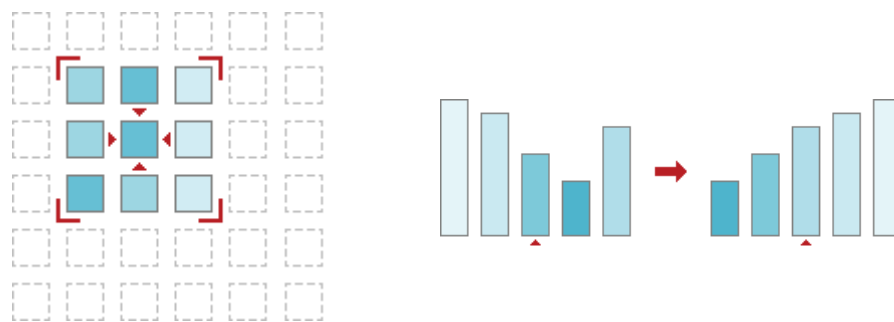


Abbildung 2–3: Quadratische Filtermaske eines Median Filters für den zwei dimensionalen Fall (links) und Sortierung der Werte nach dem Medianverfahren (rechts).

Quelle: Sergey Chernenko, www.librow.com

Wie in Abbildung 2-3 dargestellt würde der Median eines Pixels in einem zwei Dimensionalen Grauwertbildes mit einer quadratischen 3x3 Maske wie folgt berechnet:

$$\text{Median } \{(120 \ 118 \ 110)(125 \ 221 \ 105)(90 \ 101 \ 114)\} = \\ 90, 101, 105, 110, 114, 118, 120, 125, 221 = \mathbf{114}$$

Bei der Hybrid Variante des Median Filters erfolgt die Wertfindung durch mehrmalige Bildung des Median mit verschiedenen Maskenformen und erneute Bildung des Median aus den gewonnenen Ergebnissen.

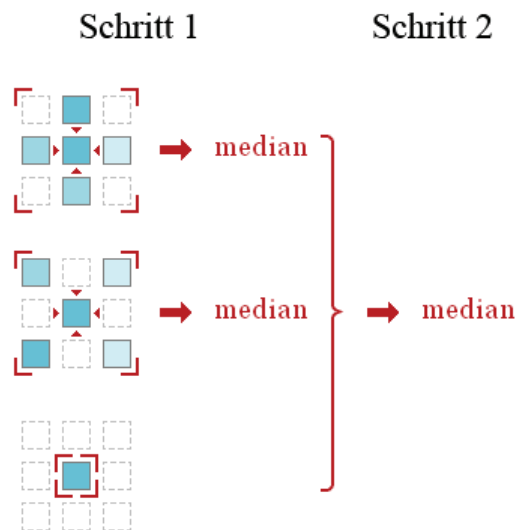


Abbildung 2-4: Die Hybrid Median Filter Maske für den zwei dimensionalen Fall.

Quelle: Sergey Chernenko, www.librow.com

In Abbildung 2-4 ist die Funktionsweise des 3D Hybrid Median Filters schematisch für den zwei dimensionalen Fall dargestellt. Wie in Schritt 1 dargestellt wird zuerst der Median für den betrachteten Pixel aus dem Median einer Kreuzmaske gebildet (Oben), anschließend findet eine X Maske für die Wertbildung Anwendung (Mitte) und optional fließt der Wert des betrachteten Pixel selbst mit ein (Unten). Im zweiten Schritt wird wiederum der Median der Werte aus Schritt 1 gebildet.

Eine Beispielrechnung würde also folglich lauten:

$$\begin{aligned} &\text{HybridMedian} \{(120 \ 118 \ 110)(125 \ 221 \ 105)(90 \ 101 \ 114)\} = \\ &[(101, 105, 125, 118, 221 = 125); (90, 110, 114, 120, 221 = 114); 221] = \\ &114, 125, 221 = \mathbf{125} \end{aligned}$$

2.2.6 DICOM Format

Das Programm benötigt für die Auswertung Angaben über das Pixel-Spacing der zu verarbeitenden Daten. Diese Information ist im Header einer Datei im DICOM Format hinterlegt.

Der durch die US-amerikanische National Electrical Manufacturers Association (NEMA) entwickelte Standard DICOM (Digital Imaging and Communications in Medicine), spezifiziert unter anderem Formate zur Speicherung und Übermittlung von Bilddaten, so dass es bei Verwendung dieses Formats möglich ist, Daten plattformunabhängig zu decodieren und zu interpretieren. Der Standard sieht vor, dass in einer Datei neben den Bilddaten noch eine Reihe zusätzlicher Informationen gespeichert werden können. Dazu gehören sowohl Informationen, die zur Interpretation der Bildinformation notwendig sind, wie zum Beispiel Angaben zur Geometrie der aufgenommenen Voxel, als auch administrative Daten, die beispielsweise die Herkunft der Bilddaten spezifizieren (DICOM 2006).

2.3 7 – Parameter – Transformation

Der Prüfkörper und die Daten des bildgebenden Systems stellen zwei Bezugssysteme mit unterschiedlicher Basis dar. Um Datenpunkte in den Systemen vergleichen zu können, muss eine Möglichkeit des Bezugssystemwechsels geschaffen werden.

Zwei verschiedene Bezugssysteme lassen sich in folgenden Punkten unterscheiden:

- Lage des Ursprungs des Referenzellipsoides
- Ausrichtung der Koordinatenachsen des Referenzellipsoides
- Maßeinheit der metrischen Koordinaten

Für den Bezugssystemwechsel ergibt sich eine Umformung mit sieben Parametern oder auch Helmert-Transformation genannt. Die Bestimmung der Transformationsparameter erfolgt mit der Methode nach Horn unter Verwendung von Quaternionen.

2.3.1 Die Helmert-Transformation

Die Helmert-Transformation (nach Friedrich Robert Helmert, 1843-1917) ist eine Koordinatentransformation für kartesische Koordinaten im dreidimensio-

nalen Raum. Sie wird häufig in der Geodäsie zur verzerrungsfreien Umrechnung von einem in ein anderes, ebenfalls dreidimensionales System genutzt.

Die Überführung eines Startsystems in ein Zielsystem erfolgt unter der Berücksichtigung von:

- 3 Translationen t_x, t_y, t_z
- 3 Rotationen R_x, R_y, R_z
- 1 Maßstabsfaktor m .

Geometrisch lässt sich die räumliche Transformation eines Punkthaufens von einem Startsystem u in ein Zielsystem x so beschreiben: Der Punkthaufen wird entlang von drei aufeinander senkrecht stehenden Richtungen durch t verschoben, um drei Achsen gedreht $R(\alpha; \beta; \gamma)$ und mit einem Maßstabsfaktor versehen. In Abbildung 2-5 ist schematisch der Übergang zwischen zwei kartesischen Systemen dargestellt.

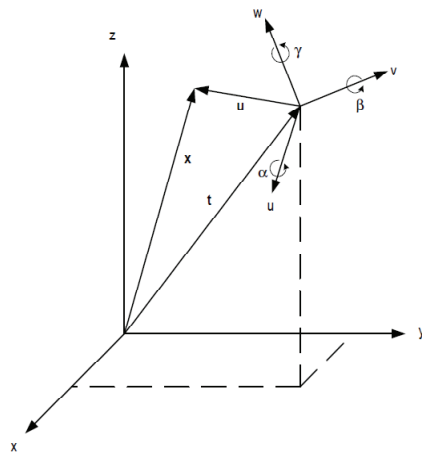


Abbildung 2–5: Beziehungen zwischen zwei räumlichen kartesischen Koordinatensystemen

Quelle: Noak 2002

Die allgemeine Helmert-Transformationsgleichung lautet:

$$Y = t + mRX$$

Formel 2-1

Transformationsgleichung

Hierbei ist der Vektor Y ein Punkt im Zielsystem und X derselbe Punkt im Startsystem.

Mit einem bekannten Parametersatz können so Koordinaten aus einem Ausgangssystem in ein Zielsystem überführt werden. Liegen keine Transformationsparameter zum Überführen von Punkten zwischen beiden Koordinatensystemen vor, so sind diese durch Stütz- bzw. Passpunkte zu schätzen. Diese Punkte besitzen Koordinateninformationen im Start- und Zielsystem.

2.3.2 Methode nach Horn

Da der Parametersatz für die Koordinatenüberführung nicht bekannt ist, müssen die Parameter über Stützstellen bestimmt werden. Aus der Geodäsie sind verschiedene Methoden bekannt. Hierzu zählen die Methode nach Bleich und Illner, die Methode der Heuristik, die Methode nach Horn sowie die Methode nach Lelgemann. Den effizientesten Algorithmus stellt dabei die Methode nach Horn dar, in der die Parameterbestimmung der räumlichen Helmert-Transformation durch ein nicht-iteratives Verfahren mittels Quaternionentransformation erfolgt (Noak 2002).

Bei der Methode nach Horn zur Parameterbestimmung handelt es sich um eine geschlossene Lösung, d.h. die 7 Parameter werden in einem Schritt berechnet und es ist nicht notwendig die Reihenfolge der Drehkonvention einzuhalten.

Zunächst sind die Koordinaten im Start- und Zielsystem auf den jeweiligen Schwerpunkt zu reduzieren.

$$x_i = X_i - \bar{x}; y_i = Y_i - \bar{y}$$

Formel 2-2

Schwerpunktreduktion

mit den Schwerpunktkoordinaten:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n X_i; \bar{y} = \frac{1}{n} \sum_{i=1}^n Y_i \quad \text{Formel 2-3}$$

Schwerpunktkoordinaten

Fasst man das Start- und Zielsystem jeweils als $n \times 3$ -Matrix S_x und S_y auf, in denen die reduzierten Punkte zeilenweise enthalten sind, so ergibt sich das Produkt M zu:

$$M = S_x^T S_y = \begin{bmatrix} m_{xx} & m_{xy} & m_{xz} \\ m_{yx} & m_{yy} & m_{yz} \\ m_{zx} & m_{zy} & m_{zz} \end{bmatrix} \quad \text{Formel 2-4}$$

Produkt der Systeme

Mit den Elementen der Matrix M kann die symmetrische Matrix N gebildet werden.

Die obere Dreiecksmatrix lautet:

$$N = \begin{bmatrix} m_{xx} + m_{yy} + m_{zz} & m_{yz} - m_{zy} & m_{zx} - m_{xz} & m_{xy} - m_{yx} \\ & m_{xx} - m_{yy} - m_{zz} & m_{xy} + m_{yx} & m_{zx} + m_{xz} \\ & & -m_{xx} + m_{yy} - m_{zz} & -m_{xy} - m_{yx} + m_{zz} \end{bmatrix} \quad \text{Formel 2-5}$$

Matrix N zur Bestimmung der Quaternion

Der größte Eigenwert mit korrespondierendem Eigenvektor ist von dieser Matrix N zu bestimmen. Dieser Eigenvektor von N ist identisch mit dem gesuchten Quaternion q , welches die Verdrehung zwischen beiden Systemen beschreibt. Eine Überführung von q in eine äquivalente Rotationsmatrix R ist [Nitschke und Knickmeyer, 2000]:

$$R = (q_0^2 - q_e^t q_e)I + 2q_e e_e^T - 2q_0^2 [q_e \times] \quad \text{Formel 2-6}$$

Rotationsmatrix aus Quaternion

Mit der so gefundenen Drehmatrix R können Maßstab m und Translation t bestimmt werden.

$$m = \sum_{i=1}^n y_i^T R x_i / \sum_{i=1}^n x_i^T x_i ; t = \bar{y} - m R \bar{x}$$

Formel 2-7

Maßstabs- und Translationsberechnung

2.3.3 Quaternionen

Wie in Kapitel 2.3.1 dargestellt muss das Startsystem u in eine parallele Lage gegenüber dem Zielsystem x durch Drehung gebracht werden. Quaternionen eignen sich besonders gut für die Beschreibung von Rotationsbewegungen und finden in der Methode nach Horn Anwendung.

Quaternionen drücken eine Rotation über einen Rotationswinkel und eine beliebige Rotationsachse aus. Für weiterführende Informationen von Rotationen mit Quaternionen sei auf Kuipers (2002) verwiesen.

Die Quaternionen (von lat. quaternion „Vierheit“) sind ein Zahlensystem ähnlich den komplexen Zahlen und eine Erweiterung der reellen Zahlen. Erdacht wurden sie 1843 von Sir William Rowan Hamilton.

Die Quaternionen entstehen aus den reellen Zahlen durch Hinzufügung dreier neuer Zahlen i, j und k. So ergibt sich in Analogie zu den komplexen Zahlen ein vierdimensionales Zahlensystem mit einem eindimensionalen Realteil und einem dreidimensionalen Imaginärteil, der auch Vektoranteil genannt wird.

Jede Quaternion lässt sich eindeutig in der Form

$$x_0 + x_1 \cdot i + x_2 \cdot j + x_3 \cdot k$$

Formel 2-8

mit den reellen Zahlen x0, x1, x3 schreiben.

3 Programmaufbau und Implementierung der Algorithmen

In diesem Kapitel erfolgt die Beschreibung, wie die im vorhergehenden Kapitel erläuterten Methoden und Klassen in der Praxis implementiert wurden. Die Gliederung orientiert sich hierbei am Programmaufbau und -ablauf.

Aufgrund des Umfangs wird der Quelltext hier nicht abgedruckt, sondern ist auf der beigelegten CD enthalten.

3.1 Entwicklungsumgebung

Das Programm ist in der Programmiersprache Java geschrieben und als Plugin für die Bildverarbeitungssoftware ImageJ angepasst. Es setzt ein installiertes Java Runtime Environment oder Development Kit voraus. Als Entwicklungsumgebung dient Eclipse IDE for Java Developers.

Das Programm ist in verschiedene Javaklassen aufgeteilt wobei die Quelldatei die Dateiendung .java, die kompilierte Javaklasse .class und das Programmpaket die Endung .jar hat. Tabelle 3.1 zeigt eine Übersicht der Klassen.

Klasse	Beschreibung
CTSphereDetect_	Hauptprogramm mit GUI. In dieser Klasse sind der 3D HybridMedian Filter sowie weitere Programmteile als Funktion integriert.
Counter3D	Hauptklasse der Objekterkennung 3D Object Counter
Object3D	Subklasse der Objekterkennung 3D Object Counter
QuaternionenTransformation	Implementierung der Methode nach Horn zur Parameterbestimmung der räumlichen Helmerttransformation

Tabelle 3-1: Die Javaklassen und ihre Funktion.

Für eine Ausführung des Programms muss die Datei CTSphereDetect.jar in den plugins-Ordner des ImageJ-Installationsordners kopiert werden. Wird ImageJ gestartet, ist zuerst ein zu analysierender Bildstapel zu öffnen. Anschließend lässt sich das Plugin über die Menüleiste des Bildverarbeitungsprogramms mit der Befehlsabfolge Plugins > CTSphereDetect starten.

Innerhalb des Programmes werden alle Daten in der Klasse *match_list* gesammelt.

3.2 Eingabedaten

Das Programm benötigt für seine Funktion Bilddaten, die als Bildstapel mit einer Bittiefe von 16bit vorliegen müssen und Angaben über das Pixel-Spacing enthalten. Im selben Verzeichnis wie die Bilddaten muss eine Textdatei mit Referenzdaten des Prüfkörpers hinterlegt sein.

Im Rahmen dieser Arbeit wurde der Prüfkörper mit einem *GALILEOS Comfort* System der Firma Sirona erfasst und als DICOM Bildstapel mit den Dimensionen 512 mal 512 Pixel auf 512 Bildebenen ausgegeben. Im Header der DICOM Daten ist ein Pixel-Spacing von 0,287 mm pro Pixel hinterlegt.

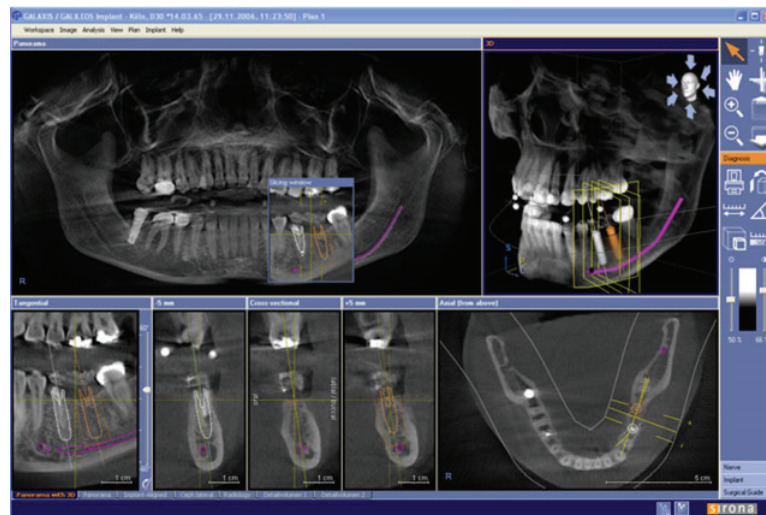


Abbildung 3–1: Sirona GALILEOS Software für dentale Röntgendiagnostik in drei Dimensionen

Quelle: www.sirona.de

3.2.1 Einlesen der Volumendaten

Mit dem ImageJ eigenen DICOM Dateireader können die Daten über das Menü als Bildstapel in den Speicher eingelesen werden. Wichtig ist die Übergabe der Kalibrierungsdaten aus dem DICOM Header an ImageJ. Diese werden für die Verarbeitung der Daten benötigt und werden über die ImageJ eigene Funktion `getCalibration()` ausgelesen.

3.2.2 Einlesen der Referenzkoordinaten

Im selben Verzeichnis wie die Bilddaten muss eine Textdatei mit dem Namen *data.txt* abgelegt werden, die die Referenzkoordinaten der im Prüfkörper eingelassenen Kugeln enthält. Die Textdatei ist nach folgendem Schema aufgebaut:

x 1 Stahlkugel	y 1 Stahlkugel	z 1 Stahlkugel
x 2 Stahlkugel	y 2 Stahlkugel	z 2 Stahlkugel
x 3 Stahlkugel	y 3 Stahlkugel	z 3 Stahlkugel
x 4 Stahlkugel	y 4 Stahlkugel	z 4 Stahlkugel
x 1 Alu-Kugel	y 1 Alu-Kugel	z 1 Alu-Kugel
x 2 Alu-Kugel	y 2 Alu-Kugel	z 2 Alu-Kugel
.	.	.
.	.	.

Wobei die Nummerierung der Kugeln des Prüfkörpers im Uhrzeigersinn und von innen nach außen ansteigt. In Abbildung 3-2 ist Nummerierung Beispielfür eine Ebene dargestellt.

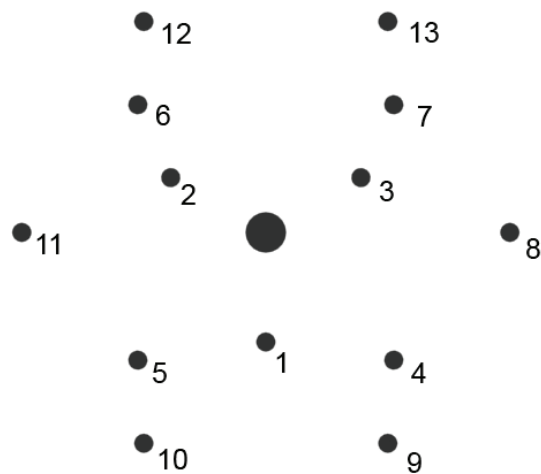


Abbildung 3–2: Nummerierungsschema der Kugeln im Prüfkörper zur Zuordnung der Koordinaten

Für den Programmablauf sind die ersten vier Einträge (Bildanker) obligatorisch, die Anzahl der Stützstellen ist variabel. Die Koordinaten in der Textdatei

werden durch die Funktion *getCoordList()* in ein zweidimensionales Array vom Typ Float eingelesen und an die Klasse *match_list.source* übergeben.

3.3 Die Benutzeroberfläche

Nach dem Start des Plugins und dem fehlerlosen Einlesen der Eingabedaten ist es erforderlich einige Prozessparameter festzulegen. Hierfür wird eine grafische Benutzeroberfläche (GUI=Graphical User Interface) verwendet. Wie in Abbildung 3-3 zu sehen, ist die GUI in drei logische Bereiche eingeteilt:

1. Schwellenwerte
2. Radien der verwendeten Kugeln und Suchtoleranz
3. Optionen

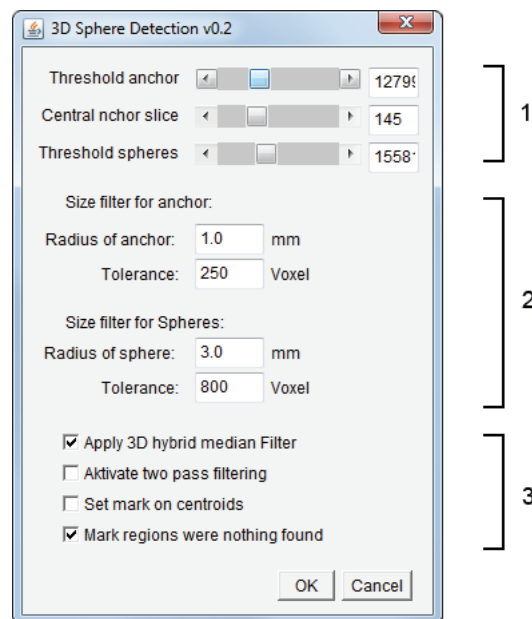


Abbildung 3–3: Die grafische Benutzeroberfläche (GUI)

Wie in Kapitel 2.2.4 erläutert arbeitet die Klasse 3D Object Counter nach dem Schwellwert-Verfahren. Mit dem Regler *Threshold anchor* wird der Schwellenwert der Bildankersuche und mit dem Regler *Central anchor slice* die ungefähre Bildebene der Anker initialisiert.

Für beide Parameter bekommt der Benutzer ein direktes visuelles Feedback (siehe Abbildung 3-4).

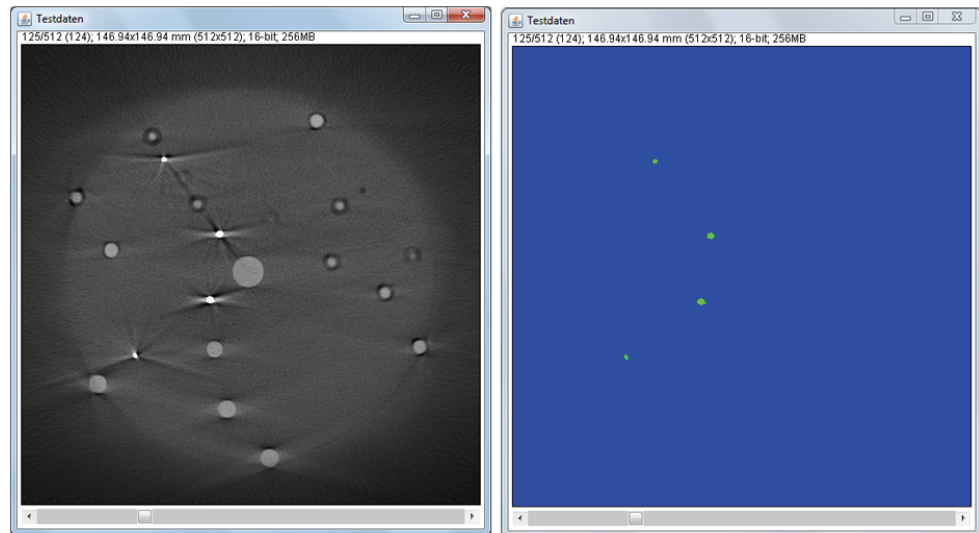


Abbildung 3-4: Bildebene mit Stahlkugeln zur Legebestimmung (links) und Visuelles Feedback der Schwellenwertanpassung (rechts)

Die im zweiten Abschnitt der GUI zu setzenden Parameter beziehen sich auf die Radien der im Prüfkörper verbauten Kugeln. In der Laufzeit des Programmes werden die Radien in Kugelvolumen mit der Einheit [Voxel] umgerechnet. Hierbei wird das in den Kalibrierungsdaten hinterlegte Pixel-Spacing berücksichtigt. Der errechnete Wert wird $\pm Tolerance$ als minimales und maximales Objektvolumen an die Klasse *Counter3D* übergeben.

Verschiedene Optionen lassen sich im dritten Abschnitt wählen. Diese beziehen sich auf die Bildaufbereitung und auf die Ausgabe der Resultate.

Im folgenden Kapitel wird näher auf die Bedeutung und Funktion der einzelnen Prozessparameter eingegangen.

3.4 Bestimmung von Orientierung und Lage des Prüfkörpers

Um Bildoptimierungs- und Objekterkennungsalgorithmen auf das volle Datenvolumen von 512^3 Voxel anzuwenden, bedarf es eines hohen Rechenaufwandes und benötigt große Mengen an Ressourcen, insbesondere Arbeitsspei-

cher (RAM). Dieser Aufwand und die Fehleranfälligkeit kann erheblich reduziert werden, wenn die Operationen nur auf relevante Bildbereiche angewandt werden. Als relevant sind Bereiche zu bezeichnen die Kugeln enthalten.

Die Positionen der Kugeln im Prüfkörper und deren Bezugssystem sind bekannt. Lage und Orientierung des Prüfkörpers relativ zum bildgebenden System sind unbekannt. Wird die räumliche Orientierung bestimmt, können die ungefähren Koordinaten der Kugeln im Datensatz berechnet werden.

Wird ein hinreichend großer Bereich um diese Koordinaten betrachtet, ist die Wahrscheinlichkeit, dass er relevante Bildelemente enthält, groß.

Zum Zwecke der Bestimmung der räumlichen Orientierung eines Körpers relativ zu einer Bezugsrichtung bedarf es fester, zuordenbarer Bezugspunkte.

Dieses Kriterium erfüllen die im Prüfkörper eingelassenen und in Abbildung 3-4 gut zu sehenden vier Bildanker.

3.4.1 Detektion der Bildanker

Es ist zu erwarten, dass der Prüfkörper innerhalb einer Toleranz aufrecht im Tomograph eingespannt wird. Da sich die Bildanker im Prüfkörper physikalisch auf einer Ebene befinden, werden auch im Datensatz die Bildanker nur eine geringe Varianz entlang der Z-Achse aufweisen. Aus diesem Grund muss nicht das gesamte Volumen nach den Bildankern durchsucht werden. Dies erhöht die Verarbeitungsgeschwindigkeit und beugt False-Positiv Ergebnissen vor.

In Kapitel 3.3 wurde mit dem Regler *Central anchor slice* der GUI, die ungefähre Lage der Bildanker auf der Z-Achse manuell bestimmt. Dieser Wert wird an die Funktion *extractRegion()* übergeben, die das Volumen der 18 vorhergehenden und nachfolgenden Bildebenen in ein eindimensionales Array vom Typ Integer schreibt.

Optional erfolgt im nächsten Schritt eine ein- oder zweifach Filterung des Teilvervolumens durch den in Kapitel 2.2.5 beschriebenen 3D Hybrid Median Filter. Dabei wird das Array an die Funktion *Hybrid3dMedianizer()* übergeben, das Er-

gebnis in ein temporäres Array geschrieben und mit *System.arraycopy()* in das Ausgangsarray zurückkopiert. Dieser letzte Schritt ist nötig, da eine einfache Variablenzuweisung bei einer Mehrfachfilterung nicht funktioniert.

Zur Detektion der eigentlichen Bildanker werden das extrahierte Volumen, der in Kapitel 3.3 durch *Threshold anchor* definierte Schwellenwert sowie das minimale und maximale Objektvolumen der Klasse *Object3D* übergeben. Werden keine, weniger oder mehr als vier Objekte zurückgegeben, wird die Laufzeit unterbrochen und eine entsprechende Fehlermeldung generiert.

Die Koordinaten der gefundenen Objekte haben als Bezugssystem das extrahierte Teilvolumen und müssen in einem abschließenden Schritt in das Bezugssystem des bildgebenden Systems transformiert werden. Dies wird durch die Funktion *putPointList()* bewerkstelligt.

3.4.2 Zuordnung der Bildanker

Die Klasse *3DCounter* gibt im Ergebnis eine Liste der gefundenen Objekte und ihrer Zentroide aus. Allerdings ist die Reihenfolge willkürlich und die detektierten Objekte müssen den Bildankern zugeordnet werden.

Die Zuordnung erfolgt in zwei Schritten:

Im ersten Schritt wird die Kollinearität der Punkte bezogen auf das Bildzentrum ermittelt. Die Funktion *LineLineIntersect()* berechnet den Schnittpunkt zweier Geraden, die wiederum durch vier Punkte definiert werden. Bei vier Bildankern ergeben sich drei Kombinationsmöglichkeiten bzw. drei Schnittpunkte. Nimmt man nun an das die Bildanker des Prüfkörpers einen Winkel von 90° bilden und der Schnittpunkt der Geraden im Zentrum des Prüfkörpers liegt, muss auch die Punktkombination von *LineLineIntersect()* korrekt sein, dessen Schnittpunkt dem Bildmittelpunkt am nächsten ist.

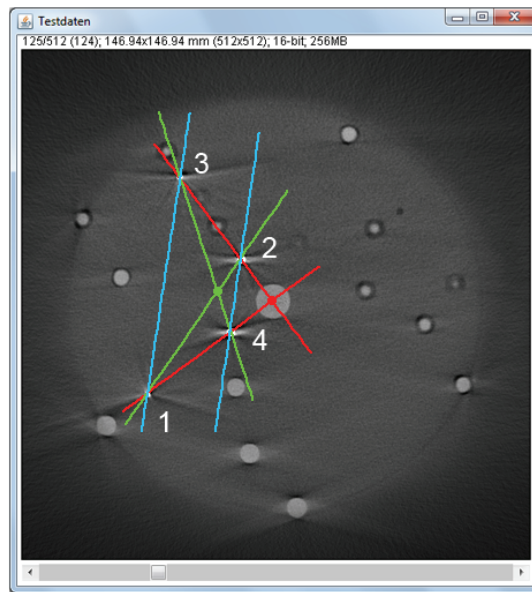


Abbildung 3-5: Zuordnung der Bildanker gemäß ihrer geometrischen Abhängigkeit.

In Abbildung 3-5 sind die drei Kombinationsmöglichkeiten der Funktion *LineLineIntersect()* visualisiert. Die Zahlen repräsentieren die Position in der Objektliste. Damit der Schnittpunkt möglichst nah am Bildzentrum liegt müssen in diesem Beispiel die Linien aus den Koordinatenpaaren (3,2) und (1,4) gebildet werden.

Im zweiten Schritt muss ermittelt werden welcher der Punkte in einem Punktpaar der Innere und welcher der Äußere ist. Hierbei erfolgt die Zuordnung durch eine erneute Berechnung des Punktabstandes zum Zentrum mittels der Funktion *dist()*.

Nach obigem Beispiel und dem in Kapitel 3.2.2 erläuterten Nummerierungsschema ist die korrekte Zuordnung:

Anker 1 \leftrightarrow Objekt 4

Anker 2 \leftrightarrow Objekt 2

Anker 3 \leftrightarrow Objekt 1

Anker 4 \leftrightarrow Objekt 3

Die Koordinaten der Ankerpunkte werden an die Klasse *match_list.target* übergeben.

3.4.3 Erste Näherung der Transformationsparameter

Die im vorangegangenen Schritt ermittelten Koordinaten der vier Bildanker in Kombination mit den dazugehörigen Referenzkoordinaten des Prüfkörpers, ermöglichen die Bestimmung der ersten Näherung für die gesuchten Transformationsparameter nach der Methode nach Horn, beschrieben in Kapitel 2.3.2. Durch Übergabe der bekannten Koordinaten des Start- und des Zielsystems an die Funktion *QuaternionenTransformation()* können fortan beliebige Koordinaten mit einer hinreichenden Genauigkeit zwischen den beiden Bezugssystemen transformiert werden und somit beide Systeme in Bezug gebracht werden.

Die Transformation wird auf die in der Klasse *match_list.source* gespeicherten Referenzkoordinaten angewendet und die neuen Koordinaten an *match_list.source_trans* übergeben.

3.5 Bestimmung der Transformationsparameter

Mit Hilfe der Bildanker konnten im vorhergehenden Schritt die Transformationsparameter näherungsweise bestimmt werden. Somit lässt sich jeder Stützstelle (Aluminium-Kugel) im Prüfkörper eine Bildkoordinate zuordnen.

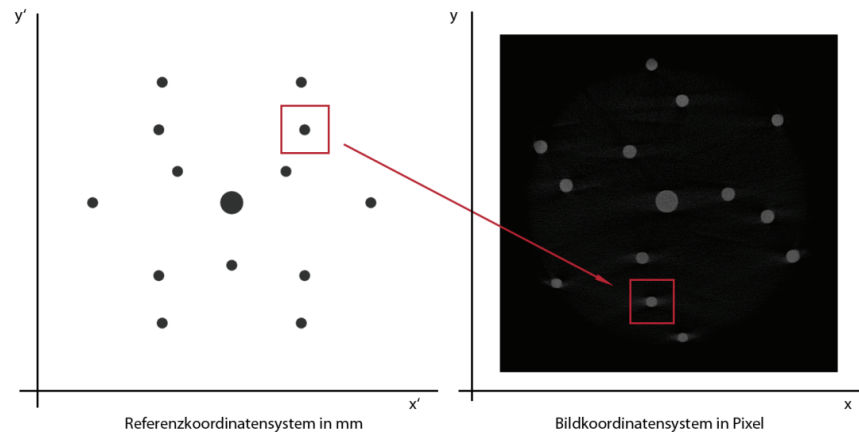


Abbildung 3–6: In das Zielsystem transformierte Referenzkoordinate

In Abbildung 3-6 ist die Beziehung zwischen Referenz- und Zielsystem dargestellt. Eine bekannte Koordinate im Referenzsystem (links) kann über die in Kapitel 3.4.3 gewonnenen Transformationsparameter in das Bezugssystem des bildgebenden Systems überführt werden (rechts).

Es ist davon auszugehen, dass die lokalen geometrischen Verzerrungen einzelner Stützstellen relativ gering sind. Somit ist mit hoher Wahrscheinlichkeit im nahen Umfeld einer transformierten Referenzkoordinate, auch eine detektierbare Struktur.

Die erste näherungsweise Bestimmung der gesuchten Transformationsparameter erfolgte auf Grundlage von vier Koordinatenpaaren. Der genauen Bestimmung der Parameter liegen je nach Detektierbarkeit bis zu 100 Koordinatenpaare zu Grunde. Man kann daher von einer Überbestimmung des Systems sprechen und somit von einer Affin – Transformation.

3.5.1 Detektion der Stützstellen

Das Verfahren zur Detektion der Stützstellen entspricht den in Kapitel 3.4 beschriebenen Schritten zu Detektion der Bildanker. Der Hauptunterschied liegt darin, dass nicht mehr ein größeres Teilvolumen auf mehrere Objekte hin untersucht wird, sondern kleine dedizierte Bereiche auf einzelne Strukturen (Kugeln).

Diese Operationen laufen in einer Schleife über alle 100 Kugeln und lassen sich wie folgt darstellen:

1. Extrahiere Teilvolumen von 50^3 Voxel um transformierte Referenzkoordinaten mit der Funktion *extractRegion()*.
2. Optional, wende 3D Hybrid Median Filter zu kantenerhaltenden Rauschreduzierung mit der Funktion *Hybrid3dMedianizer()* auf das Teilvolumen an
3. Übergebe das Teilvolumen der Klasse *3D Counter()* mit einem definierten minimalen und maximalen Objektvolumen
4. Wenn ein Objekt gefunden wird, übergebe dessen Koordinaten an die Klasse *match_list.target*.
5. Wird kein Objekt gefunden, erniedrige Schwellenwert in 5% Schritten bis Schwellenwert/2 erreicht ist oder ein Objekt gefunden wird
6. Bezugssystemwechsel des extrahierten Teilvolumens durch *putPointList()*.

Die Koordinaten der gefundenen Stützstellen werden an die Klasse *match_list.target* übergeben.

3.5.2 Behandlung von falsch positiv Ergebnissen

In Bildbereichen mit sehr niedrigem Signal-Rausch Verhältnis oder an den Randbereichen des Aufnahmevervolumens kann es zu falsch positiv Ergebnissen kommen. Das heißt, dass Bildelemente als Stützstelle detektiert werden die keine sind.

Um diesem Umstand Rechnung zu tragen, erfolgt nach der Stützstellendetektion eine Plausibilitätsprüfung. Dabei wird, wie in Abbildung 3-7 dargestellt, mit

Hilfe der in Kapitel 3.4.3 bestimmten näherungsweisen Transformationsparameter, die im Bildraum detektierte Stützstelle in das Referenzsystem überführt.

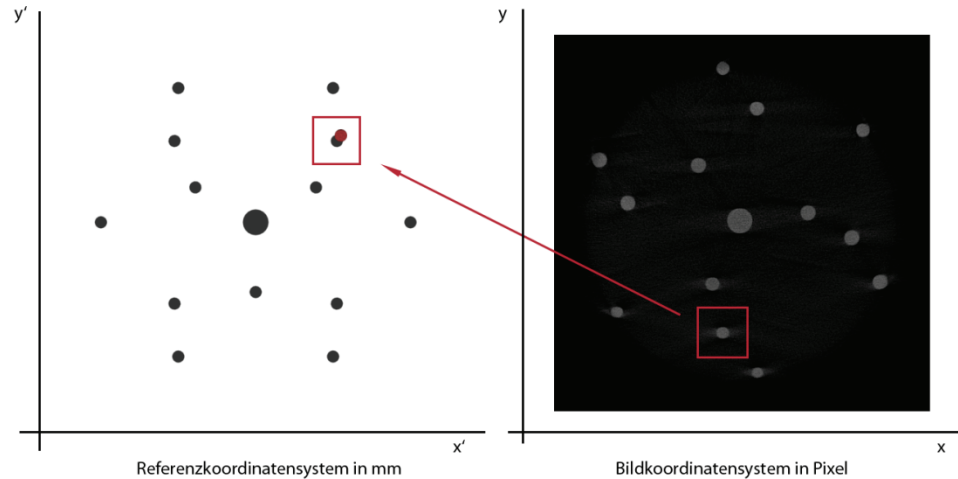


Abbildung 3–7: Transformation einer Stützstelle in das Referenzsystem

Anschließend errechnet die Funktion *dist()* die Abweichung des gemessenen Objektzentrums von der Referenzkoordinate. Übersteigt die Abweichung einen vordefinierten Wert, wird die Messung verworfen. Derzeit wird die Messung bei einer Abweichung $> 2,5$ mm verworfen und geht nicht mehr in das Ergebnis ein. Dieser Wert ist statisch im Quelltext hinterlegt. In einer höheren Programmversion sollte der Wert anforderungsbezogen veränderbar sein.

3.5.3 Bestimmung des lokalen geometrischen Fehlers

Aus den vorangegangenen Schritten liegt nun eine hohe Zahl an systembeschreibenden Stützstellen für eine erneute genaue Bestimmung der Helmert-Transformationsparameter t , m und R vor.

Wie bei der ersten näherungsweisen Bestimmung, werden die Koordinatenpaare des Referenz- und Bildsystems der Funktion *QuaternionenTransformation()* übergeben und damit mit der Methode nach Horn die Parameter bestimmt.

Die Transformation wird auf die in der Klasse *match_list.target* gespeicherten Bildkoordinaten angewendet und die neuen Koordinaten werden an *match_list.target_trans* übergeben.

Aus den Koordinaten die in *match_list.source* und *match_list.target_trans* hinterlegt sind, kann der geometrische lokale Fehler durch Differenzbildung ermittelt werden.

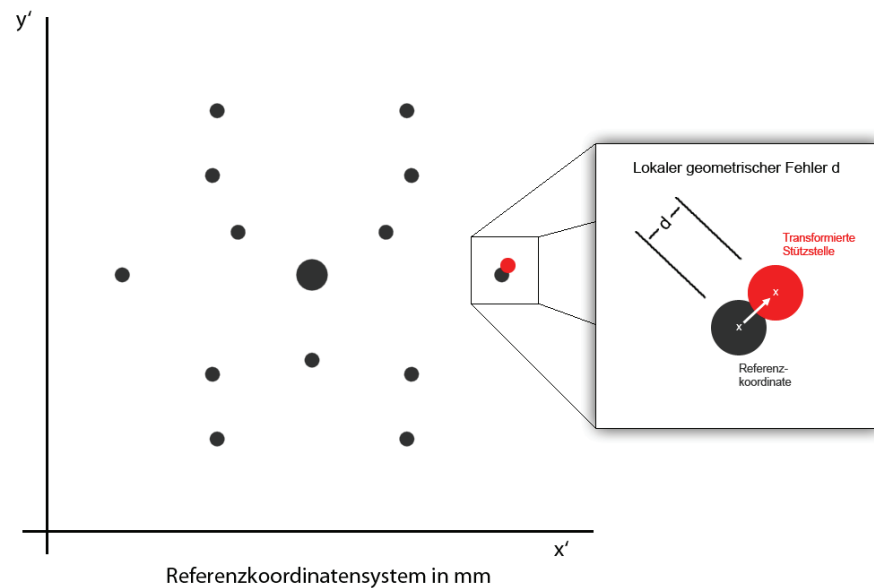


Abbildung 3–8: Lokaler geometrischer Fehler des bildgebenden Systems

Wie in Abbildung 3-8 dargestellt, bildet der Differenzvektor von Referenzpunkt und transformierter Stützstelle, die lokale Verzerrung in Richtung und Stärke ab.

3.6 Ausgabe der Ergebnisse

In der Klasse *match_list()* sind alle während der Laufzeit des Programmes gewonnenen Daten hinterlegt. Um die Daten besser bewerten und interpretieren zu können, wurde mit Hilfe der ImageJ eigenen Funktion *ResultsTable()* eine tabellarische Aufbereitung der Ergebnisse realisiert.

Sphere #	Found	X	Y	Z	X'	Y'	Z'	Deviation	Deviation XY	Deviation XZ	Deviation YZ	Threshold	Xt	Yt	Zt	X''	Y''	Z''
1	1	15	0	39	15.130	0.304	39.089	0.343	0.331	0.157	0.317	3071	210.021	282.506	125.942	210.267	283.625	125.591
2	2	0	-15	39	0.489	-15.085	39.358	0.612	0.497	0.606	0.368	3071	222.247	209.653	128.209	220.694	210.360	126.917
3	3	45	0	39	45.049	1.036	39.229	1.062	1.037	0.234	1.061	3071	124.909	343.119	123.420	126.857	346.143	122.539
4	4	0	-45	39	1.111	-44.990	39.311	1.154	1.111	1.154	0.311	3071	161.587	124.560	130.220	158.455	126.789	129.033
5	5	1	15.590	-9	63	15.636	-9.089	62.981	0.102	0.100	0.050	1955	190.858	255.689	42.920	190.542	255.533	42.972
6	6	1	-15.590	-9	63	-15.584	-8.975	62.956	0.051	0.026	0.044	1955	279.318	192.691	45.541	279.351	192.774	45.674
7	7	1	0	18	63	0.017	18.160	62.939	0.172	0.161	0.063	1839	289.681	300.773	42.420	289.959	301.272	42.601
8	8	1	0	-23	57	0.472	-23.138	57.053	0.494	0.492	0.475	1723	206.603	185.100	66.063	204.982	185.650	65.833
9	9	1	-19.920	11.500	57	-20.213	11.074	56.997	0.517	0.517	0.293	1839	332.875	242.710	65.425	332.854	240.910	65.472
10	10	1	19.920	11.500	57	19.825	11.931	56.889	0.455	0.441	0.445	1723	219.847	323.204	62.075	220.980	324.254	62.427
11	11	1	33.780	-19.500	57	34.125	-18.931	57.011	0.666	0.666	0.345	1839	117.844	263.279	62.989	118.005	265.591	62.869
12	12	1	-33.770	-19.500	57	-33.467	-20.108	56.994	0.679	0.679	0.304	1723	309.487	126.797	68.668	307.402	125.675	68.690
13	13	1	0	39	57	-0.748	39.073	56.937	0.754	0.751	0.750	1723	331.966	360.958	61.906	334.241	359.673	62.166
14	14	1	15.590	-9	43	15.650	-9.173	43.146	0.234	0.183	0.158	1839	190.267	257.757	112.566	189.747	257.373	112.055
15	15	1	-15.590	-9	43	-15.613	-9.005	43.023	0.033	0.023	0.033	1723	278.727	194.759	115.188	278.784	194.693	115.098
16	16	1	0	18	43	0.039	18.065	43.082	0.112	0.076	0.091	1723	289.090	302.841	112.067	289.116	303.102	111.762
17	17	1	36	0	43	36.062	-0.214	43.139	0.263	0.223	0.152	1839	150.561	324.522	110.247	149.945	324.033	109.761
18	18	1	0	-36	43	0.051	-36.115	43.139	0.188	0.126	0.148	1839	179.903	149.675	115.688	179.523	149.427	115.195
19	19	1	-36	0	43	-36.039	0.008	42.960	0.056	0.039	0.055	1839	354.829	179.050	116.300	354.963	178.992	116.431
20	20	1	0	36	43	0.053	36.072	43.066	0.111	0.089	0.084	1839	325.486	353.897	110.860	325.491	354.210	110.611
21	21	1	45.900	-26.500	43	45.827	-26.690	43.132	0.243	0.204	0.151	1839	68.891	269.359	111.192	68.701	268.661	110.739
22	22	1	0	-53	43	-0.010	-52.995	43.163	0.164	0.011	0.164	1839	145.530	101.455	116.827	145.561	101.418	116.248
23	23	1	-45.900	-26.500	43	-45.798	-26.434	43.042	0.129	0.122	0.111	1839	329.333	83.882	118.909	329.184	84.255	118.740
24	24	1	-45.900	26.500	43	-45.730	26.642	42.965	0.224	0.222	0.174	1839	436.498	234.212	115.356	436.320	234.961	115.444
25	25	1	0	53	43	0.095	52.975	43.011	0.099	0.099	0.096	1839	359.859	402.116	109.720	359.549	402.250	109.663
26	26	1	45.900	26.500	43	45.821	26.411	43.080	0.143	0.119	0.112	1839	176.056	419.689	107.638	176.093	419.285	107.361
27	27	1	0	-27	37	0.635	-27.100	37.103	0.651	0.643	0.644	1723	197.924	175.823	135.978	195.916	176.805	135.564
28	28	1	-23.380	13.500	37	-23.622	12.995	36.986	0.560	0.560	0.242	1723	346.144	243.460	135.228	345.618	241.539	135.323
29	29	1	23.380	13.500	37	23.134	13.964	37.108	0.536	0.525	0.269	1723	213.484	337.936	131.297	215.118	338.751	130.901
30	30	1	31.470	-31.470	37	31.796	-31.119	37.185	0.513	0.479	0.374	1839	98.604	226.727	133.632	99.380	228.359	132.928
31	31	1	-31.470	-31.470	37	-30.998	-31.921	37.122	0.664	0.653	0.467	1839	278.168	99.561	138.924	275.924	99.207	138.482

Abbildung 3–9: Tabellarische Ausgabe der Ergebnisse

Die in Abbildung 3-9 dargestellte Ergebnistabelle teilt sich in drei logische Bereiche auf:

Auf der linken Seite werden Kugelindex, Status und Prüfkörperbezogene Koordinaten erfasst. Im mittleren Bereich ist der lokale geometrische Fehler, in Stärke und Richtung, aufgelistet. Rechts sind alle, das bildgebende System betreffenden Koordinaten verzeichnet.

Zur Kontrolle werden, zusätzlich zu der Ergebnistabelle, in einem neuen Bildstapel die mittleren Bildebenen aller untersuchten Teilvolumina hinterlegt.

4 Ergebnisse

In diesem Kapitel werden die Ergebnisse des Programms betrachtet. Dazu werden ein virtueller Datensatz mit bekannten Größen, sowie eine unter realen Bedingungen bei der Firma Sirona entstandener Datensatz des Prüfkörpers verwendet.

Wird ein Datensatz mehrmals mit gleichen Einstellgrößen des Programms ausgewertet, so ergeben sich immer identische Werte. Damit kann die Wiederholgenauigkeit (precision) der automatischen Auswertung als so genau beschrieben werden, dass sie in der Fehlerbetrachtung keine Rolle spielt. Beide Datensätze sind auf der dieser Arbeit beiliegenden CD enthalten.

4.1 Rauschreduktion

Im Rahmen dieser Arbeit wurde in das Programm ein 3D Hybrid Medianfilter zur kantenerhaltenden Rauschminderung implementiert. In Abbildung 4-1 wurden über ein 16bit Ausgangsbild, normalverteiltes Rauschen mit unterschiedlicher Standardabweichung gelegt. Es ist gut zu sehen wie durch die Filterung das Rauschen reduziert wird, die ursprüngliche Struktur aber erhalten bleibt.

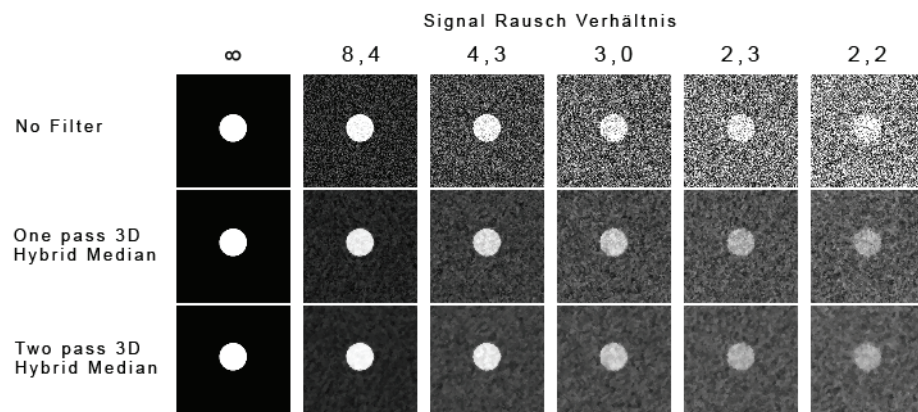


Abbildung 4-1: Gegenüberstellung der Filterergebnisse des 3D Hybrid Median Filters in Abhängigkeit des SNR

Um die Performanz des Filters hinsichtlich einer Empfindlichkeitssteigerung bei der Detektion zu überprüfen, wurden vom virtuellen Datensatz Kopien mit

unterschiedlichem Signal-Rausch Verhältnis (SNR) angefertigt. Dazu wurde der Bildstapel mit einem normalverteiltem Rauschen mit einer Standardabweichung zwischen 0k bis 20k überlagert. Anschließend wurden die Datensätze mit den Programmeinstellungen keine, einfache und zweifache Filterung analysiert.

In Abbildung 4-2 wird deutlich, dass die Filterung Einfluss einerseits auf die Verarbeitungsdauer hat, andererseits die Algorithmen bei höherem Rauschen noch funktionieren.

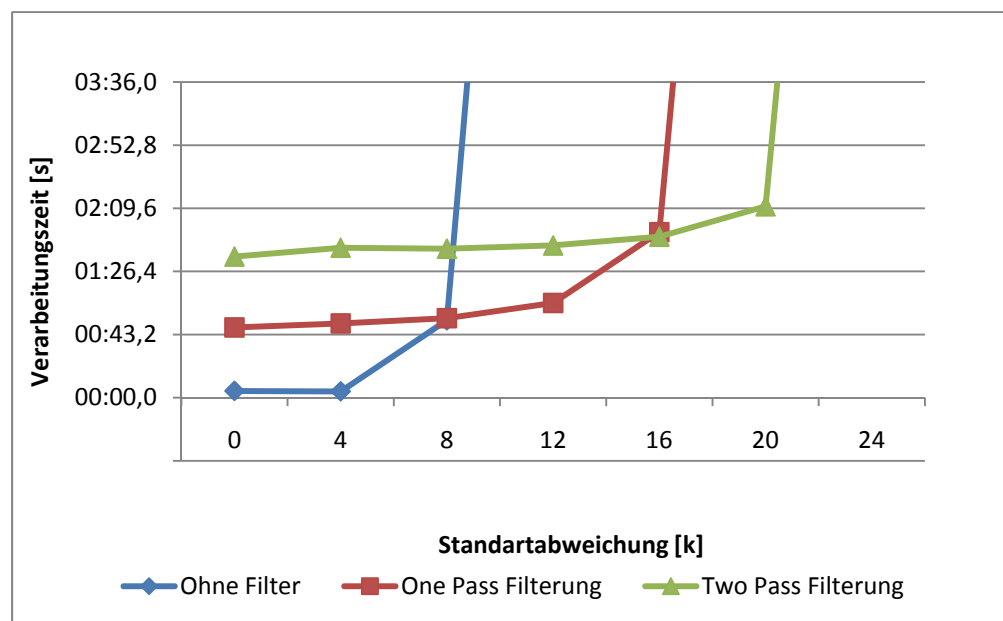


Abbildung 4–2: Verarbeitungszeit in Abhängigkeit des Rauschens mit unterschiedlicher Standardabweichung

Ohne Filterung (Blau) war eine Verarbeitung der Daten bis zu einer Standardabweichung von 8K möglich. Bei höherem Rauschen ging die Verarbeitungsdauer gegen unendlich. Bei Anwendung einer einfachen Filterung war eine Verarbeitung des Virtuellen Datensatzes bis zu einer Standardabweichung von 16K, bei zweifacher Filterung bis zu 20 K möglich. Ob und in welchem Maße die Filterung einen Einfluss auf die Messergebnisse hat ist Gegenstand der Fehlerbetrachtung.

4.2 Fehlerbetrachtung

Grundsätzlich treten bei jeder Messung Umstände auf, die dafür verantwortlich sind, dass der gemessene Wert vom tatsächlichen Wert abweicht. Um die Abweichung, des in dieser Arbeit entwickelten Verfahrens zu bestimmen, werden die Resultate der im vorhergehenden Abschnitt beschriebenen virtuellen Datensätze hinsichtlich ihres mittleren Fehlers untersucht.

4.2.1 Virtueller Datensatz

Der Virtuelle Datensatz enthält vier Bildanker und 27 gleichmäßig verteilte Stützstellen. Das Pixel-Spacing beträgt 0,287 mm/px. Er ist als fehlerfrei zu betrachten.

Nr.	Standardabweichung des Rauschens und Filterart	Mittlerer Fehler [mm]	Mittelwertabw. [mm]	Maximaler Fehler [mm]	Gültige Stützstellen
1	0K kein Filter	0,056	0,016	0,078	27
2	0k einfache Filterung	0,056	0,016	0,077	27
3	0K zweifache Filterung	0,056	0,016	0,077	27
4	4K kein Filter	0,057	0,017	0,081	27
5	4K einfache Filterung	0,056	0,017	0,079	27
6	4K zweifache Filterung	0,056	0,017	0,079	27
7	8K kein Filter	0,056	0,015	0,079	27
8	8K einfache Filterung	0,060	0,016	0,086	27
9	8K zweifache Filterung	0,061	0,017	0,088	27
10	12K einfache Filterung	0,060	0,015	0,091	27
11	12K zweifache Filterung	0,058	0,014	0,091	27
12	16K einfache Filterung	0,063	0,017	0,114	23
13	16K zweifache Filterung	0,066	0,018	0,116	25
14	20K zweifache Filterung	0,076	0,018	0,139	22

Tabelle 4-1: Mittlerer Fehler der Kugeldetektion im virtuellen Datensatz in Abhängigkeit des Rauschens und der Filterart

Der Tabelle 4-1 ist die Messunsicherheit (accuracy) der Kugeldetektion im virtuellen Datensatz in Abhängigkeit des Rauschens und der Filterart, sowie die jeweilige Mittelwertabweichung und der maximale Fehler zu entnehmen. Die

dieser Tabelle zugrundeliegenden Messwerte sind in Form einer Microsoft Excel Datei auf der dieser Arbeit beiliegenden CD enthalten.

In Abbildung 4-3 sind die in Tabelle 4-1 aufgelisteten Ergebnisse grafisch dargestellt.

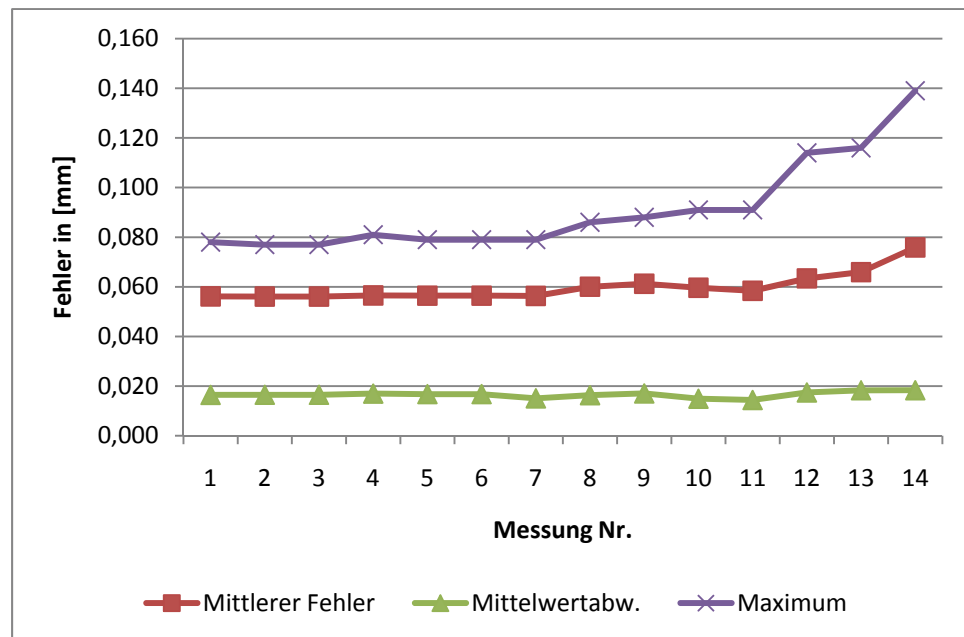


Abbildung 4–3: Grafische Darstellung der Ergebnisse aus Tabelle 4-1

Mit ansteigendem Rauschen, steigt der mittlere Fehler, wobei die Filterart einen vernachlässigbaren Einfluss hat. Zwischen Filterart und Anzahl der gefundenen Stützstellen, in Abhängigkeit des Rauschens, ist ein Zusammenhang zu beobachten. Die Mittelwertabweichung bleibt konstant.

Daraus ergibt sich durch erneute Mittelung unter Berücksichtigung der Anzahl gültiger Messergebnisse ein durchschnittlicher maximaler Systemfehler des Verfahrens von 0,08 mm bzw. 0,28 Pixel.

In Tabelle 4-2 wird der vektorielle Anteil des mittleren Fehlers in die Bildebenen XY, XZ und YZ aufgeschlüsselt. Als Datengrundlage dient der virtuelle Datensatz ohne Rauschen und ohne Filterung. Die Werte weichen bei anderen virtuellen Datensätzen nur marginal ab.

	Gesamt	XY	XZ	YZ
Mittlerer Fehler [mm]	0,056	0,032	0,050	0,049
Mittelwertabweichung [mm]	0,016	0,010	0,018	0,019
Minimum [mm]	0,003	0,003	0,003	0,002
Maximum [mm]	0,078	0,053	0,075	0,076

Tabelle 4-2: Vektorieller Anteil des mittleren systemischen Fehlers in den Bildebenen XY, XZ und YZ

Der Fehler in den vertikalen Bildebenen ist geringfügig größer als in der horizontalen Bildebene. Ursachen hierfür könnten im Algorithmus der Klasse *3DCounter()* liegen. Um dies zu verifizieren bedarf es weiterer Versuche, die den Umfang der vorgelegten Arbeit übersteigen.

4.2.2 Reeller Datensatz

Eine exemplarische Auswertung des realen Datensatzes ergibt folgende Ergebnisse:

Nr.	Filterart	Mittlerer Fehler [mm]	Mittelwertabw. [mm]	Maximaler Fehler [mm]	Gültige Stützstellen
1	Keine Filterung	0,473	0,253	1,310	76
2	Einfache Filterung	0,464	0,268	1,334	78
3	Zweifache Filterung	0,471	0,269	1,345	81

Tabelle 4-3: Mittlerer Fehler der Kugeldetektion im realen Datensatz in Abhängigkeit der Filterart

Wie beim virtuellen Datensatz zeigt die Filterung einen vernachlässigbaren Einfluss auf den Wert des mittleren Fehlers. Die Qualität des Ergebnisses wird jedoch durch eine höhere Zahl an verwertbaren Stützstellen, insbesondere im Randbereich des Prüfkörpers verbessert.

Unter Berücksichtigung der in Kapitel 4.4.1 gewonnen Erkenntnisse und der Fertigungstoleranz des Prüfkörpers, ergibt sich für das bildgebende System ein durchschnittlicher mittlerer Abbildungsfehler von 0,47 mm mit einer Unsicherheit von $\pm 0,18$ mm.

5 Diskussion

Im folgenden Kapitel werden die Ergebnisse der vorliegenden Arbeit diskutiert. Dabei liegt ein besonderes Augenmerk auf der Anwendbarkeit des Verfahrens in der Bestrahlungstherapie.

Die Strahlentherapie erfordert heute durch die hohen applizierten Dosen eine sehr genaue Ortsbestimmung. Mit geeigneten Algorithmen und leistungsfähigen Planungscomputern können Dosisberechnungen mit sehr hoher Genauigkeit erreicht werden. Zudem liegen die Toleranzen der Einstellungsgenauigkeit der modernen Linearbeschleuniger in der Hochpräzisionsbestrahlung, bei $\pm 0,5$ mm.

In Kapitel 4 wurde dargelegt, das mit dem in dieser Arbeit entwickelten Programm, Strukturelemente und deren Zentroide in einem Prüfkörper detektiert werden können. Dabei wurde eine mittlere systemische Unsicherheit von 0,08 mm ermittelt.

Die mittlere systemische Unsicherheit wurde auf Basis eines virtuellen Datensatzes berechnet. Ein reeller Prüfkörper ist fertigungsbedingt fehlerbehaftet. Der im Rahmen dieser Arbeit entwickelte und verwendete Prüfkörper weist eine Fertigungstoleranz von 0,1 mm auf. Dieser Wert sollte bei einer kommerziellen Fertigung des Prüfkörpers so weit erniedrigt werden, dass er bei der Bestimmung der Messunsicherheit des Prüfverfahrens keine große Rolle mehr spielt und die Messunsicherheit des Messverfahrens im wesentlichen durch die Messunsicherheit der Algorithmen bestimmt wird. Ein Wert von etwa 0,04 mm wäre ausreichend. Damit ergäbe sich eine systemische Messunsicherheit von etwa 0,1 mm. Dies muss bei der Betrachtung und Beurteilung der Leistungsfähigkeit des Systems berücksichtigt werden.

Im vorliegenden Fall ergäbe sich somit eine Gesamtunsicherheit von etwa $\pm 0,10$ mm bei der Bestimmung des lokalen geometrischen Fehlers eines CT- oder DVT-Systems.

Eine hinreichend genaue Bestimmung der räumlichen Unsicherheit eines realen Systems innerhalb der für die Hochpräzisionsbestrahlungstherapie benötig-

ten Toleranzen, ist demzufolge möglich. Damit wird sowohl für die konventionelle Strahlentherapie (Radiotherapie, RT) wie auch für Hochdosisverfahren eine verbesserte Bestrahlungsplanung ermöglicht. Neben diesem Aspekt sei darauf hingewiesen, dass es möglich ist durch die Auswahl geeigneter Materialien für einen Prüfkörper diesen sowohl in radiologischen Verfahren, wie CT oder DVT zu verwenden, wie auch in Magnetresonanzverfahren einzusetzen. Damit könnte die Fusion unterschiedlicher bildgebender Verfahren ebenfalls erheblich verbessert werden, da beide Verfahren sich auf eine systemische Unsicherheit abstützen können, die mit konventionellen Fusionsverfahren nicht erreicht wird, da die systemische Unsicherheit des jeweiligen Einzelverfahrens additiv in die Unsicherheit des fusionierten Bilddatensatzes eingeht.

Zur Verifizierung dieser Aussagen müssen weitergehende Untersuchungen durchgeführt werden.

Neben dem Aspekt der Strahlentherapie, kann eine verbesserte räumliche Auflösung auch für die Planung operativer Eingriffe sinnvoll sein, da z.B. gerade bei operativen Eingriffen am Schädel sehr hohe Genauigkeitsanforderungen einzuhalten sind.

6 Zusammenfassung und Ausblick

Ziel dieser Arbeit war es eine Software zu entwickeln, die Strukturelemente in Volumendaten automatisch detektiert und anhand von Referenzdaten den lokalen geometrischen Fehler ermittelt. Die besondere Herausforderung bei der Entwicklung bestand in dem Entwurf eines Verfahrens mit möglichst geringen Fehlertoleranzen.

Dazu wurde ein Prüfkörper entwickelt und gefertigt. Anschließend wurden Aufnahmen mit einer DVT Einrichtung des Herstellers Sirona erstellt, die als Datengrundlage für einen ersten Test des Programms dienen. Nachdem die Bilddaten mit ImageJ eingelesen sind, wird zunächst die Orientierung und Lage des Prüfkörpers relativ zum bildgebenden System mit Hilfe von Bildankern ermittelt. Dabei wird über eine GUI die ungefähre Position der Bildanker im Datensatz vorgegeben und anschließend ein Teilvolumen extrahiert. Optional kann mit einem speziellen Median Filter das Bildrauschen vermindert werden. Mit einem Algorithmus zur Objektdetektion werden die Bildanker im Teilvolumen lokalisiert. Im Anschluss werden mit der Methode nach Horn näherungsweise Transformationsparameter ermittelt, um die Bezugssysteme des Prüfkörpers und des bildgebenden Systems in Relation setzen zu können. Die näherungsweise bestimmten Transformationsparameter ermöglichen eine Abbildung der Referenzkoordinaten in den Bilddatensatz und somit eine gezielte Eingrenzung der Suchbereiche nach Strukturelementen. Nach einer Plausibilitätsprüfung der Ergebnisse, werden diese für eine genaue Bestimmung der Transformationsparameter herangezogen. Es ist nun eine Bestimmung der lokalen geometrischen Abweichung vom Sollwert möglich.

Es konnte festgestellt werden, daß eine Anwendung dieses Verfahrens in der Bestrahlungstherapie möglich ist, da die Meßunsicherheit des Verfahrens hinreichend gering ist und zu einer Verbesserung der Bestrahlungsplanung beitragen kann.

Um eine noch höhere Genauigkeit und Robustheit des Verfahrens zu erreichen, sind weitere Verbesserungen insbesondere bei der Herstellung des Prüfkörpers und ggf. im Bereich der Objekterkennung und Bildbearbeitung denk-

bar. Ein Algorithmus zur Objekterkennung der gezielt nach Kugelstrukturen sucht könnte die Qualität des Ergebnisses noch weiter verbessern. Im Bereich der Bildbearbeitung wäre die Verwendung eines nicht linearen Gaussfilters denkbar, der eine verbesserte Kantenerhaltung bei der Rauschunterdrückung verspricht (Bauer 2005). Ein weiterer Ansatzpunkt für Verbesserungen ist die Konstruktion und Fertigung des Prüfkörpers. Bei einer Fertigungstoleranz von 10^{-3} mm würde das Ergebnis signifikant verbessert werden. Darüber hinaus ist über die Verwendung von kleineren Kugeln nachzudenken, um die Stützstellendichte zu erhöhen.

Zusätzlich sind weitere Untersuchungen erforderlich, um eine bessere Abschätzung über den Nutzen des Verfahrens in der Praxis treffen zu können.

A Anhang

A.1	Abbildungsverzeichnis	38
A.2	Tabellenverzeichnis	39
A.3	Formelverzeichnis.....	39
A.4	Inhalt der CD	39
A.5	Literaturverzeichnis	40

A.1 Abbildungsverzeichnis

Abbildung 2–1:	Schematischer Aufbau (links) und mit ImageJ realisierte Volumendarstellung der Daten des mit DVT erfassten Prüfkörpers (rechts).....	4
Abbildung 2–2:	Verfahren der Objektzuordnung durch das ImageJ Plugin „3D Object Counter“	7
Abbildung 2–3:	Quadratische Filtermaske eines Median Filters für den zwei dimensional Fall (links) und Sortierung der Werte nach dem Medianverfahren (rechts).....	8
Abbildung 2–4:	Die Hybrid Median Filter Maske für den zwei dimensional Fall.....	9
Abbildung 2–5:	Beziehungen zwischen zwei räumlichen kartesischen Koordinatensystemen.....	11
Abbildung 3–1:	Sirona GALILEOS Software für dentale Röntgendiagnostik in drei Dimensionen	16
Abbildung 3–2:	Nummerierungsschema der Kugeln im Prüfkörper zur Zuordnung der Koordinaten	17
Abbildung 3–3:	Die grafische Benutzeroberfläche (GUI)	18
Abbildung 3–4:	Bildebene mit Stahlkugeln zur Lagebestimmung (links) und Visuelles Feedback der Schwellenwertanpassung (rechts)...	19
Abbildung 3–5:	Zuordnung der Bildanker gemäß ihrer geometrischen Abhängigkeit	22
Abbildung 3–6:	In das Zielsystem transformierte Referenzkoordinate	24
Abbildung 3–7:	Transformation einer Stützstelle in das Referenzsystem	26
Abbildung 3–8:	Lokaler geometrischer Fehler des bildgebenden Systems.....	27
Abbildung 3–9:	Tabellarische Ausgabe der Ergebnisse.....	28
Abbildung 4–1:	Gegenüberstellung der Filterergebnisse des 3D Hybrid Median Filters in Abhängigkeit des SNR	29
Abbildung 4–2:	Verarbeitungszeit in Abhängigkeit des Rauschens mit unterschiedlicher Standardabweichung	30
Abbildung 4–3:	Grafische Darstellung der Ergebnisse aus Tabelle 4-1	32

A.2 Tabellenverzeichnis

Tabelle 2-1: Im Prüfkörper verbaute Materialien unterschiedlicher Elektronendichte.....	3
Tabelle 3-1: Die Javaklassen und ihre Funktion.....	15
Tabelle 4-1: Mittlerer Fehler der Kugeldetektion im virtuellen Datensatz in Abhängigkeit des Rauschens und der Filterart.....	31
Tabelle 4-2: Vektorieller Anteil des mittleren systemischen Fehlers in den Bildebenen XY, XZ und YZ.....	33
Tabelle 4-3: Mittlerer Fehler der Kugeldetektion im realen Datensatz in Abhängigkeit der Filterart.....	33

A.3 Formelverzeichnis

Formel 2-1	12
Formel 2-2	12
Formel 2-3	13
Formel 2-4	13
Formel 2-5	13
Formel 2-6	13
Formel 2-7	14
Formel 2-8	14

A.4 Inhalt der CD

- Programmdateien
- Quellcode
- Datensätze
- Ergebnistabellen
- Diplomarbeit

A.5 Literaturverzeichnis

- Berthold K.P. Horn, B.K.P. (1987):** *Closed-form solution of absolute orientation using unit quaternions*. Erscheinungsort: Journal of the Optical Society of America
- H. Fröhlich, H.Körner (2001):** *Geodätische Koordinatentransformationen - Ein Leit-faden*. Erscheinungsort: Selbstverlag
- Jack B. Kuipers (2002):** *Quaternions and Rotation Sequences - A Primer with Appli-cations to Orbits, Aerospace and Virtual Reality*. Erscheinungsort: Princeton University Press
- S. Bolte & F. P. Cordelières (2006):** *A guided tour into subcellular colocalization analysis in light microscopy*. Erscheinungsort: Journal of Microscopy, Vo-lume 224, Issue 3: 213-232
- F. P. Cordelières (2009):** *The 3D object counter plugin*. Erscheinungsort: ImageJ wiki webpage <http://imagejdocu.tudor.lu/>
- M. Noak (2002):** *Verschiedene Verfahren zur Bestimmung der Parameter einer Helmertrtransformation*. Erscheinungsort: Seminararbeit, Technischen Uni-versität Berlin
- C. P. Mauer (2007):** *3d Hybrid Median Filter*.Erscheinungsort: <http://rsbweb.nih.gov/ij/plugins/hybrid3dmedian.html>
- C. Bauer (2005):** *Extraktion und Modellerstellung von Gefäßbäumen aus medizinischen Volumendaten*. Erscheinungsort: Fraunhofer-Institut FIT
- D. Lammers & S. Wachenfeld (2003):** *Objekterkennung in Bilddaten*. Erschei-nungsort: Seminar: „Unterstützung von Landminendetektion durch Bildauswertungsverfahren und Robotereinsatz“, Universität Münster, Institut für Informatik.

Eidesstattliche Erklärung

Ich versichere hiermit, die vorgelegte Arbeit in dem gemeldeten Zeitraum ohne fremde Hilfe verfasst und mich keiner anderen als der angegebenen Hilfsmittel und Quellen bedient zu haben.

Köln, den 25.02.2010

Christian Uphoff

Sperrvermerk

Die Einsicht in die vorgelegte Arbeit ist bis zum 32. 13. 2009 gesperrt.

Weitergabeerklärung

Ich erkläre hiermit mein Einverständnis, dass das vorliegende Exemplar meiner Diplomarbeit oder eine Kopie hiervon für wissenschaftliche Zwecke verwendet werden darf.

Köln, den 25.02.2010

Christian Uphoff